## APPENDIX A.

## EXEMPLARY GRAMMAR FILE SPECIFICATION

A grammar file is made up of rules. Rules preferably have the following form:

```
Definition
RULE_NAME:
Definition;
END
```

The "RULE_NAME:" is the unique name of the rule. The "RULE_NAME:" can be made of any characters A-Z, a-z, 0-9 or '_'. It can only start with the characters A-Z or a-z and it must end with a ':'.

"END" marks the end of a rule.

The "Definitions;" is the definition of the rule. Rule definitions must end in a semi-colon (;). Rule definitions can be made of the following components:

P(XXXX)
A reference to a specific prompt where 'XXXX' is the four digit key to the specific prompt. For example, P(0001) references the prompt, "I'm sorry you are having trouble," in the US English grammar.
<RULE_POINTER>
A reference to another previously defined rule. "RULE_POINTER" must match a name of a rule already defined in the grammar.
{PROMPT}
A special place holder for a dynamically created prompt. A dynamically created prompt could be a user's name or greeting.

A simple example of a rule is:

```
HOLD_FOR_EXTENSION:
P(9007)P(5132)<DIGIT>;
END
```

There are five types of Rules:

Simple
A simple rule can have any number of prompts and at most one other rule. Below are a few examples of simple rules.

Rule for saying a phrase like, "Please hold for extension 200."

```
HOLD_FOR_EXTENSION:
P(9007)P(5132)<DIGIT>;
END
```

Rule for saying a phrase like, "Extension 200 is unavailable."
EXTENSION_UNAVAILABLE:
P(5135)<DIGIT>P(5136);
END

Rule for saying a phrase like, "I'm sorry, extension 200 does not answer."
SORRY_EXTENSION_DOES_NOT_ANSWER:
P(5131)P(5132)<DIGIT>P(5139);
END

Complex

A complex rule is just like a simple rule, but it can have any number of other rules. Below are a few examples of complex rules.

Rule for saying a phrase like, "Message received at 1:00pm on Tuesday, June 22."
TIME_STAMP:
P(5171)P(5114)<TIME>P(9001)<DATE>;
END

Rule for saying a phrase like, "The system will try to notify you from 8:00 AM until 5:00 PM, calling every 15 minutes."
NOTIFY_SCHEDULE:
P(5168)<TIME>P(5169)<TIME>P(5170)<DURATION_MINUTES>;
END

Prompt

A prompt rule is simply a '{PROMPT}' component. Below are a few examples of rules that contain a prompt rule.

Rule for saying a phrase like, "Your recorded name is 'Chris Johnson'."
RECORDED_NAME:
P(5167){PROMPT};
END

Rule for saying a phrase like, "Please hold for 'Fred Sarkissian'."
HOLD_FOR_NAME:
P(9000){PROMPT}P(5141);
END

Index

An index rule is a rule with multiple rule definitions. An example of an index rule is:
ORDINAL:
[1] P(5057);
[2] P(5058);
[3] P(5059);

```
[4] P(5060);
[5] P(5061);
END
```

An index rule can have a special modifier called 'RANGE', which is a short cut for repeating a similar rule across multiple indexes. For example:

```
RANGE_TEST:
[1] P(5057);
[2] P(5058);
[3] P(5059);
[4] P(5060) RANGE 10;
END
```

This rule would replicate the rule definition of 'P(5060)' up till and including index 10. It is the same as the rule:

```
RANGE_TEST:
[1] P(5057);
[2] P(5058);
[3] P(5059);
[4] P(5060);
[5] P(5060);
[6] P(5060);
[7] P(5060);
[8] P(5060);
[9] P(5060);
[10] P(5060);
END
```

Digit

A digit rule is a special case of an index rule. It has the name 'DIGIT' and it is a rule that provides digit prompts for other rules. A digit rule does this by automatically using all the digits passed as parameters. Normally, if the parameter list of "274" was passed to an ordinary index rule then the first parameter '2' would be stripped off and the rest of the parameters "74" would be passed to the rule found at the index 2. A digit rule, by contrast, will continually call itself as long as the first parameter is a number, appending the results of the last call of the digit rule to the previous. Thus a parameter list of "274" would result in the prompts from index 2, 7 and 4 being appended together in that order. By definition there can only be one digit rule per grammar and indexes of digit rules should not contain Rule Pointers. Example of usage is:

```
HOLD_FOR_EXTENSION:
P(9007)P(5132)<DIGIT>;
END
```

An exemplary digit rule is:

```
DIGIT:
[0] P(5026);
[1] P(5027);
[2] P(5028);
[3] P(5029);
[4] P(5030);
[5] P(5031);
[6] P(5032);
[7] P(5033);
[8] P(5034);
[9] P(5035);
END
```